

# **ANEXOS**

## **CUESTIONARIO. -**

- I. ¿QUIEN MANEJA EL SISTEMA?**
- II. ¿COMO CONTROLAN LAS FICHAS O RESERVAS DE TURNO?**
- III.COMO CONTROLAN LAS ASISTENCIAS DEL MEDICO Y OPERADORES?**
- V. ¿COMO CONTROLAN EL COBRO DE CADA CONSULTA?**
- VI. ¿COMO REGISTRAN LOS HISTORIALES MEDICOS DE CADA PACIENTE?**
- VII. ¿COMO CONTROLAR LAS DERIVACIONES?**
- VIII. ¿COMO SABER SI LAS HABITACIONES DE EMERGENCIAS ESTAN DISPONIBLES?**

## **RESPUETAS**

R1.- Solo maneja la secretaria para registrar a los pacientes.

R2.- Controlan en hojas o en un cuaderno la secretaria contratada.

R3.- Los médicos y operadores tienen una entrada a sus consultorios y registran con firmas en un cuaderno.

R4.- Administran o se hace cargo un medico designado quien con aprobación de los demás médicos y operador hacen contrataciones de operadores y se organizan con reuniones.

R5.- La secretaria es la encargada de cobrar a los pacientes después de su atención respectiva y ella es la que se hace cargo de controlar la Re consulta.

R6.- Registran con folder respectivos para cada persona o en algunos casos como familia.

R7.- Controlan mandando en hojas de derivación que tienen los médicos con su firma y sello.

R8.- Se controla eso por Llamadas a la clínica directas hacia la secretaria para avisar si están disponibles para atención de emergencia o para informar que serán atendidas.

**Dr. Alfonso Ernesto Escobar Fernandez**  
Cardiología Pediatra  
Av. Dr. Emilio Aguero y Calle Dr. Carlos Gonzalez Buitrago  
Centro Médico Siglo XXI, No. 43, Colonia Miraflores, San Salvador  
TELÉFONO: 2228-9902 • 7943-4754

**FACTURA**  
Nº 0211  
REGISTRO No. 155266-7  
NIT: 5614-250963-001-7

Fecha: 26 de Julio - 2013

Cliente: [Redacted]  
Dirección: [Redacted]

Venta a Cuenta

CANT.	DESCRIPCION	PRECIO UNITARIO	VMS NO	PSE	VENTAS AFECTAS
1-	Capsula medica				230.00
1-	Electrocardiograma				225.00
1-	Ecocardiograma				220.00

Sumas: \$175.00  
(-) IVA RETENIDO  
SUB-TOTAL  
VENTA NO SUJETA  
VENTA EXENTA

Entregado por: [Redacted]  
Nombre: [Redacted]  
DUI: [Redacted]  
Firma: [Redacted]

**Clinica Modelo**  
Cirurgia Dentista - CRO 90.000  
Clínica General - Estética  
Especializado en Pedodontia

Recebi do(a) Sr(a): \_\_\_\_\_

C.P.F.: \_\_\_\_\_ a quantia de R\$ \_\_\_\_\_

Referente à: \_\_\_\_\_

S. B. do Campo, \_\_\_\_\_ de \_\_\_\_\_ 20 \_\_\_\_\_

Rua nonononono, 00 - Bairro - 00000-000 - S. Bernardo do Campo  
drigraph@hotmail.com Fone/Fax: (11) 2356-4717

**Hospital Regional "SAN JUAN DE DIOS"**  
Tarija - Bolivia

**INFORME ESTADÍSTICO DE INGRESOS Y EGRESOS DE HOSPITALIZACIÓN**

Nombre: \_\_\_\_\_ Fecha de ingreso: \_\_\_\_\_ Hora de ingreso: \_\_\_\_\_

Sexo: \_\_\_\_\_ Edad: \_\_\_\_\_ Fecha y hora de egreso: \_\_\_\_\_

Ocupación: \_\_\_\_\_ Puesto de ingreso: \_\_\_\_\_

Residencia Actual: \_\_\_\_\_ Departamento: \_\_\_\_\_ Provincia: \_\_\_\_\_ Localidad: \_\_\_\_\_ Calle N° \_\_\_\_\_

Residencia Temporal: \_\_\_\_\_ Departamento: \_\_\_\_\_ Provincia: \_\_\_\_\_ Localidad: \_\_\_\_\_ Calle N° \_\_\_\_\_

Número de veces hospitalizado en este mismo año: \_\_\_\_\_ Hospitalización por: \_\_\_\_\_

Fecha de ingreso: \_\_\_\_\_ Al servicio de: \_\_\_\_\_ A la sala de: \_\_\_\_\_

Fecha de primer traslado: \_\_\_\_\_ Al servicio de: \_\_\_\_\_ A la sala de: \_\_\_\_\_

Fecha de alta: \_\_\_\_\_ Hora: \_\_\_\_\_ Del servicio de: \_\_\_\_\_ De la sala de: \_\_\_\_\_

Número de días de estadía: \_\_\_\_\_ Intervención quirúrgica - especificar: \_\_\_\_\_ SI NO

**DIAGNÓSTICO DE EGRESO:**  
Principal 1.-  
Otros 2.-  
3.-  
4.-

Causa interna en caso de accidente

Autopsia SI NO Causa de alta: \_\_\_\_\_ Condición al egresar: \_\_\_\_\_

Mejorado  
No mejorado  
Fallecido

Para maternidad Sexo: \_\_\_\_\_ Condición al nacer: \_\_\_\_\_ Peso en gr: \_\_\_\_\_

Recién nacido Masculino Muerto  
Femenino Vivo

**HOSPITAL REGIONAL "SAN JUAN DE DIOS"**  
Tarija - Bolivia

**ANAMNESIS**

AP. PATERNO  
AP. MATERNO

NOMBRES  
N° DE H. CL.  
N° DE CAMA

1. Motivo de Hospitalización; 2. Enfermedad Actual; Síntomas y Signos de inicio y Evolución, Medicación y tratamientos recibidos; 3. Antecedentes personales: Particulares, Desarrollo Psicosomático, vacunas, hábitos; 4. Antecedentes patológicos: Enfermedades de infancia, Enfermedades de los órganos de los sistemas, alergias; 5. Antecedentes patológicos familiares.

FUENTE: Colegio Médico de Cochabamba  
GRÁFICO: Los Tiempos / Wilson Cahuaya

# **Especificación de Requisitos según la** **NORMA IEEE-830**

**SISTEMA AUTOMATIZADO  
DE ADMINISTRACION Y CONTROL MEDICO  
CLINICA MEDICA - "EL BUEN SAMARITANO"**



## **1 Introducción**

Este documento es una especificación de Requisitos Software para el Sistema Web para la administración y control médico de la Clínica Médica “El Buen Samaritano”.

El centro médico “El Buen Samaritano” empezó a operar en la sociedad tarijeña desde el año 2013, La misma que centra sus servicios en la atención a la población de la ciudad de Tarija.

En un principio el centro médico empezó a funcionar con pocos pacientes y haciendo su reserva de citas médicas en hojas de papel, pero con el transcurso del tiempo las historias y registros realizados se fueron incrementando. Debido a esto se planteó la realización de un sistema software para la administración y control médico de la Clínica Médica “El Buen Samaritano”.

En resumen, se trata de desarrollar una aplicación para facilitar la recepción de citas médicas, así como ayudar a su gestión, de una forma sencilla y clara para los usuarios y el administrador de la aplicación.

Esta especificación se ha estructurado basándose en las directrices dadas por el estándar “IEEE Recommended Practice for Software Requirements Specification ANSI/IEEE 830 1998”

### **1.1 Propósito**

El objeto de la especificación es definir de manera clara y precisa todas las funcionalidades y restricciones que se desea construir. El documento va dirigido a las personas encargadas del desarrollo e implementación del sistema, al propietario del comercial, al tutor del proyecto.

Este documento será el canal de comunicación entre las partes implicadas, tomando parte en su desarrollo miembros de cada parte.

Esta especificación está sujeta a revisiones, que se recogerán por medio de sucesivas versiones del documento, hasta alcanzar su aprobación por las partes involucradas. Una vez aprobado servirá de base para la construcción del nuevo sistema.

### **1.2 Alcance**

El motor que impulsa el desarrollo del sistema es la evidencia de una creciente competencia en el rubro, además de la complicada y morosa tarea de realizar un registro de pacientes y de citas medicas, por otra parte, se tendrá un registro de pacientes y registro de citas que realiza cada encargado de citas. La situación de partida es una en la que no existe un sistema informático que automatice lo antes mencionado y gestione una serie de tareas que se consideran necesarias para la óptima administración de la clínica.

El futuro sistema deberá ser capaz de registrar nuevos pacientes y actualizar el historial de los ya existentes en los registros de la clínica, además de generar un registro de citas médicas para facilitar así la atención y los beneficios de los pacientes, además de realizar los reportes correspondientes a las citas y historiales.

El número de usuarios que tendrán acceso a la aplicación será mediano, del orden de 10-20 personas. El modo de acceso será a través de un sistema web, previa validación de usuario y contraseña, habrá dos niveles de usuarios, aquellos que puedan manejar el registro de los pacientes y el administrador que registra a los usuarios de la clínica que se encargarán de las diferentes tareas.

### 1.3 Personal Involucrado

<b>Nombre</b>	Ulises Yamil Posadas Alanez
<b>Rol</b>	Analista, diseñador y programador
<b>Categoría profesional</b>	Estudiante
<b>Responsabilidades</b>	Análisis de información, diseñador y programador del sistema
<b>Información de contacto</b>	<a href="mailto:ulyss.19@gmail.com">ulyss.19@gmail.com</a>

### 1.4 Definiciones, acrónimos y abreviaturas

- **Usuario:** Persona autorizada a usar las funcionalidades del sistema
- **Administrador:** Persona con acceso, no solo a las funcionalidades, también a la base de datos
- **BD:** Base de datos.

### 1.5 Referencias

IEEE Recommended Practice for Software Requirements Specification ANSI/IEEE STD 830, 1998

### 1.6 Visión General del Documento

Este documento consta de tres secciones. Esta sección es la introducción y proporciona una visión general de la ERS. En la sección 2 se da una descripción general del sistema, con el fin de conocer las principales funciones que debe realizar, los datos asociados y los factores, restricciones, supuestos y dependencias que afectan al desarrollo, sin entrar en excesivo detalle. En la sección tres se definen con más detalle los requisitos que debe satisfacer el sistema.

## **2 Descripción general**

### **2.1 Perspectiva del producto**

El objetivo es que tanto los usuarios como el administrador puedan acceder al sistema desde un navegador web. Tanto los usuarios como el administrador deberán tener facilidad de acceso a las funcionalidades del sistema. Lo ideal es que no dependan nunca de terceras personas para hacer pequeños cambios en la BD.

El sistema, en principio, no interactuará con ningún otro sistema informático. Sería deseable, en un plazo medio, que interactuase con alguna aplicación que facilite el envío de mensajes, mail, sms a los pacientes que hayan registrado en el centro médico.

En la parte de programación, la actualización de inventarios se busca que el sistema actualice automáticamente el registro al momento de registrar un paciente nuevo o el aumento de especialidades en uno ya existente, en el momento de efectuar esta acción el sistema deberá informar al usuario que la acción se realizó con éxito, para luego poder realizar las atenciones médicas, las atenciones médicas se registran solo al momento de que el paciente esté presente en la clínica, esto debido a las políticas de la clínica; el centro de salud no realiza registros por web. El sistema será capaz de realizar reportes diarios de las atenciones médicas. El sistema almacenará toda la información correspondiente a los pacientes que realicen una atención, así poder realizar un reporte de beneficiarios, como también el registro de las garantías de los pacientes atendidos.

Para todos estos procesos lo primordial es que la base de datos cuente con los mínimos datos como ser, la actualización del historial, las atenciones en el centro médico, el registro correspondiente de cada paciente y usuario, se guardan datos como el nombre, carnet de identidad, teléfono, etc. Con estos datos se busca respuestas con un alto grado de exactitud ya que se trata de un sistema basado en métodos heurísticos que los expertos manejan en forma empírica.

### **2.2 Funcionalidades del producto**

En términos generales, el sistema deberá proporcionar soporte a las siguientes tareas:

- Actualización y consulta de información sobre los registros almacenados en la Clínica Médica.
- Consulta de información sobre las atenciones médicas del centro médico.
- Consulta de información sobre los pacientes y empleados del centro médico.
- Consulta de Información sobre los registros de citas médicas de la clínica.
- Relación bidireccional con usuarios.



- Obtención de estadísticas, como ser mayor atención en cuanto al mes.

Por otro lado, los aspectos no-funcionales del sistema serán los siguientes:

- Sistema web, fácil de usar
- Simplicidad y facilidad de manejo

A continuación, se describirán con más detalle estos aspectos y como deberían ser soportados por el sistema

### **2.2.1 • Actualización y consulta de información sobre inventario del centro de salud “EL BUEN SAMARITANO”**

La base de datos incorporará una serie de atributos que enriquezcan a través de la actualización de la información sobre los usuarios, pacientes y cantidad de cada uno de los historiales médicos almacenados en la base de datos.

El objetivo es permitir consultas avanzadas que permiten obtener, por ejemplo, cuantos pacientes se atendieron al mes en el centro de salud, o cual fue el médico que más atención tubo en el mes en el centro médico o cuantas especialidades existen en el centro de salud.

### **2.2.2 Consulta de información sobre las atenciones médicas del centro de salud “EL BUEN SAMARITANO”**

La base de datos incorporará una serie de atributos que enriquezcan la información sobre las citas médicas que fueron registradas en el centro de salud, cuantas atenciones se realizaron por día semana o mes, cuantos pacientes nuevos fueron registrados.

El sistema permitirá que las citas médicas se registren con fichas para así poder controlar las citas médicas realizadas en el centro de salud.

### **2.2.3 Consulta de información sobre los pacientes y empleados del centro de salud.**

La base de datos incorporará una serie de atributos que enriquezcan la información sobre los datos de los usuarios de la clínica y pacientes, esto por razón de seguridad y protección de la empresa, el sistema podrá realizar un reporte en donde se pueda revisar datos de los usuarios del centro de salud y de los pacientes con datos como la dirección de su domicilio, teléfono y también los datos personales.

### **2.2.4 Relación bidireccional con usuarios**

Sería conveniente que el sistema, sobre el BD de usuarios de la clínica y pacientes, permita la construcción dinámica y flexible de lista de cada usuario de la clínica que realizó registro de citas médicas reflejado en una lista a partir de una

serie de atributos de las citas y usuarios de la clínica. Esto facilitará el control del registro de citas médicas y registro de pacientes y los responsables de las mismas en caso de haber reclamos o para simple control.

Idealmente, este proceso de construcción dinámica de listas que muestra las citas de cada usuario de la clínica, debería ser lo más automatizado posible.

### 2.2.5 Obtención de estadísticas

Periódicamente se deseará extraer información que permita evaluar tendencias (reportes de atenciones médicas, reportes de pacientes y otros análisis relevantes). En principio no parece necesario que estos análisis los haga el sistema, y bastaría con que los datos se puedan extraer de la BD para luego ser analizados con algún paquete estadístico. Dicha extracción de datos podría hacerla el administrador, cuando sea necesario, mediante consultas SQL.

### 2.3 Características de los usuarios

<b>Tipo de usuario</b>	Usuario anónimo
<b>Formación</b>	Secundaria
<b>Habilidades</b>	Conocimientos de PC básicos y navegación por internet
<b>Actividades</b>	Este usuario no tendrá acceso al sistema por ahora necesita registro.

<b>Tipo de usuario</b>	Pacientes
<b>Formación</b>	Secundaria
<b>Habilidades</b>	Conocimientos de PC básicos y navegación por internet
<b>Actividades</b>	Este usuario no tendrá acceso al sistema por ahora necesita registro.

<b>Tipo de usuario</b>	Administrador
<b>Formación</b>	Secundaria
<b>Habilidades</b>	Conocimientos de PC básicos
<b>Actividades</b>	Altas, bajas y modificaciones de especialidades, médicos, pacientes, usuarios de la clínica; además será encargado de revisar las atenciones,

<b>Tipo de usuario</b>	Medico
<b>Formación</b>	Profesional
<b>Habilidades</b>	Conocimientos de PC básicos e internet

<b>Actividades</b>	Altas, bajas y modificaciones de diagnósticos médicos, pacientes;
--------------------	---

<b>Tipo de usuario</b>	Secretaria
<b>Formación</b>	Profesional o Título Técnico
<b>Habilidades</b>	Conocimientos de PC básicos e internet
<b>Actividades</b>	Altas, Bajas y además será encargado de revisar las atenciones,

El sistema de información deberá ofrecer una interfaz de usuario estándar, fácil de aprender y sencillo de manejar. Lo deseable sería que un usuario nuevo, con el único prerequisite de ser una persona acostumbrada al uso de un navegador web, se familiarizarse con el sistema en menos de una hora.

## 2.4 Restricciones

Es importante destacar que el sistema ayudará a los pacientes sirviendo como un asistente para su registro de atenciones médicas, toda la información requerida para efectuar las atenciones deberá ser brindada por el paciente. Por ejemplo, si no se sabe el precio de alguna especialidad es imposible realizar la cita médica, si el sistema no cuenta con la información mínima acerca de los médicos disponibles no podrá realizar la recepción de cita médica. Es por tal motivo que la base de datos deberá ser actualizada una vez al mes para obtener buenos resultados, logrando que la clínica medica “EL BUEN SAMARITANO” cumpla con las metas.

Dado que el sistema implementará la política y los procesos actualmente vigentes es de esperar que futuros cambios en los modos de trabajo o en las políticas, ejerzan cierto impacto sobre el sistema.

Otra restricción importante es la naturaleza de la infraestructura del software a utilizar, pues siempre será preferible utilizar software libre. La opción PostgreSQL será probablemente la opción final, por la amplia base de desarrolladores que existen con conocimiento de dicho entorno de desarrollo.

Asimismo, este documento considerará los requisitos de seguridad derivados de la ley de Protección de datos actualmente vigente.

Otra de las restricciones del sistema que se considera importante es que no se controlará la parte contable de la clínica medica, tampoco se contempla la gestión de plantillas de los empleados.

El sistema será diseñado según los requerimientos de la clínica médica “EL BUEN SAMARITANO”

El sistema requiere capacitación y mantenimiento.

## **2.5 Suposiciones y dependencias**

### **2.5.1 Suposiciones**

En el documento se expresan los requisitos en términos de lo que el sistema debe proporcionar a los usuarios que acceden a él para consultar. No obstante, el sistema no proporcionará nada útil a menos que haya alguien que introduzca los datos. Se asumirá, por tanto que:

- El administrador será el responsable de realizar una carga inicial de datos, a partir de los datos actuales de Profesiones, especialidades, médicos y pacientes.
- Los usuarios con acceso al sistema tienen un conocimiento básico en el uso de un computador.
- Información actualizada
- Obtención de información ágil y rápida
- Reducción y detección de errores en cuanto a la información.
- No existirá pérdida de información de acuerdo al control y seguridad que requiera el software

Se asume que los requisitos descritos en este documento son estables una vez que sea aprobado por la gerencia del centro médico “EL BUEN SAMARITANO”. A partir de ese momento, cualquier petición de cambios en esta especificación deberá ser aprobada por todas las partes, previo análisis del impacto que dicha petición de cambio supondrá para el sistema, para la BD o para los usuarios.

### **2.5.2 Dependencias**

En el día a día, el sistema funcionará autónomamente, sin necesidad de comunicarse con otros sistemas similares, por lo que no hay dependencias respecto de otros sistemas.

## **2.6 Evolución previsible del sistema**

En un futuro se tiene previsto incorporar al sistema la parte contable del centro de salud “EL BUEN SAMARITANO”.

### **3 Requisitos específicos**

En este apartado se presentan los requisitos funcionales que deberán ser satisfechos por el sistema. Todos los requisitos aquí expuestos son esenciales, es decir, no sería aceptable un sistema que no satisfaga alguno de los requisitos aquí planteados.

Los requisitos expuestos en esta sección deberán cumplir los criterios de “testeabilidad” y trazabilidad, permitiendo de esta manera, probar fácilmente el cumplimiento de los mismos por parte del sistema y acompañar la evolución natural del sistema ante los cambios en los requerimientos.

Cada requisito tiene asignada una prioridad de implementación, que puede ser alta, media o baja. La prioridad alta es para aquellas funciones cuya ausencia no sería tolerable en el futuro sistema. La prioridad media es para aquellas que necesitan más discusión. Finalmente, la prioridad baja es para las funciones que no son necesarias ahora, pero deberán incorporarse en el medio-largo plazo.

#### **3.1 Requerimientos comunes de las interfaces**

##### **3.1.1 Interfaces de usuario**

La interfaz de usuario deberá ser un paciente web de acuerdo a la definición de RWC del W3C, permitiendo entre otras cosas, la obtención y procesamiento asincrónico de datos; el cálculo y procesamiento en el paciente y una interfaz intuitiva que facilite la navegación.

En la interfaz de usuario de nuestra aplicación inicialmente se accederá a una página principal donde en el centro aparecerán todas las categorías de nuestras especialidades medicas.

Dispondremos de un apartado que nos permitirá identificarnos, y en el caso de que el usuario no sea paiente, le permitirá realizar su respectivo registro enviando un formulario y procediendo a su alta.

Según los permisos de cada tipo de usuario, se podrán acceder a las distintas opciones del sistema.

Se dispondrá de un apartado que nos facilitará la búsqueda de productos, introduciendo una palabra buscará los artículos que la incluyan.

Por último dispondremos de logout para poder desconectarnos de la sesión.

### Pantalla Principal

CLINICA MEDICA - "EL BUEN  
SAMARITANO"

### Adicionar Usuario

Adicionar Usuarios

Carnet de Identidad	Nombre
<input type="text"/>	<input type="text"/>
Primer Apellido	Segundo Apellido
<input type="text"/>	<input type="text"/>
Telefono	Direccion
<input type="text"/>	<input type="text"/>
Email	Foto
<input type="text"/>	<input type="text"/>

Examinar

Cancelar

Guardar

## Gestion Citas

**Gestion**

100 x 30

Perfil

Gestion Citas

Buscar

☐ Inactivos ☒ Activos

AGREGAR

MODIFICAR

ELIMINAR

Menu

Personas ▼

Sistema ▼

Organizacion ▼

Profesiones

Salas

Seguro Social

Dosificacion

Citas

Acciones ▼

## Adicionar Pacientes

**Adicionar Pacientes**

Carnet de Identidad

Nombre

Primer Apellido

Segundo Apellido

Fecha de Nacimiento

Direccion

Email

Sexo

☐ Femenino ☒ Masculino

Telefono

Seguro Social

Numero de Seguro Social

Grupo Sanguineo

NIT

Nombre de NIT

Observacion

Cancelar

Guardar

### Gestión Consultas

The screenshot shows a web application window titled 'Gestion'. On the left is a sidebar menu with options: 'Menu', 'Personas', 'Sistema', 'Organización', 'Acciones', 'Consultas' (highlighted), and 'Informes'. The main content area is titled 'Gestion Consultas' and includes a search bar with a 'Buscar' button. Below the search bar are radio buttons for 'Inactivos', 'Activos' (selected), 'Pendientes' (selected), 'Realizadas', 'Faltantes', and 'Postergadas'. There are three buttons: 'AGREGAR', 'MODIFICAR', and 'ELIMINAR'. A large empty box is present below the radio buttons. The top right of the window has standard window controls and a 'Perfil' button.

#### 3.1.2 Interfaces de hardware

- Computadora con procesador Pentium IV o superior
- 1 Gb de memoria RAM

Los clientes necesitarán una máquina que dispongan de acceso a internet, actualmente cualquier equipo que soporte un navegador web, con una tarjeta de red o un modem.

La máquina que realizará la función de servidora almacenará toda la información del catálogo de productos, además de la base de datos con la información de los usuarios y clientes registrados. Esta máquina necesitará una conexión a internet de banda ancha, para así poder asistir las peticiones de los clientes.

#### 3.1.3 Interfaces de software

La aplicación va a ser desarrollada bajo el sistema operativo Windows 7, sobre este sistema operativo instalaremos nuestro sistema, que incorporará Apache como servidor, postgresSQL como gestor de base de datos y como lenguaje de implementación JAVA.



Para el desarrollo del interfaz web utilizaremos el editor de texto *Sublime* que nos ayudará con el HTML, como también las hojas de estilo mediante *CSS3*.  
Por último, la plataforma de software Eclipse que nos ayudará a programar.

Los clientes que accedan a nuestra web no será necesario que dispongan de un sistema operativo concreto, el navegador deberá estar actualizado a las últimas versiones para evitar visualizaciones erróneas o falta de plug-in o complementos.

### **3.1.4 Interfaces de comunicación**

- Dominio de internet

Los protocolos de comunicación van a ser TCP/IP entre los clientes y la web y la comunicación entre los navegadores y el servidor HTTP.

Los clientes y usuarios podrán conectarse a través de internet.

### **3.2 Requisitos funcionales**

El sistema deberá mantener la integridad referencial de los datos en todo momento. Los datos ingresados deberán ser validados de acuerdo a los dominios indicados en el diccionario de datos al momento de ser ingresados.

A continuación se describen las funcionalidades que debe proporcionar el sistema.

#### **Actualización y consultas de registro de pacientes**

##### **Req (01) Registro. Pacientes Prioridad: Alta**

**Introducción.** - El sistema actualizará los pacientes con los nuevos datos y atributos correspondientes a cada registro.

**Entrada.** - Paciente seleccionado.

**Proceso.**- La aplicación abrirá el correspondiente vínculo que hace referencia a la actualización y consulta de pacientes registrados en la clínica medica.

**Salida.**- En la pantalla aparecerán todos los detalles de la petición referente a la categoría seleccionada.

##### **Req (02) consulta.recuperación.simple Prioridad: Alta**

**Introducción.-** La recuperación y presentación de los datos de un paciente, en su forma más sencilla, se podrá hacer a partir de su código de registro.

**Entrada.-** Código de registro de paciente.

**Proceso.-** La consulta se realizará a la base de datos procesando internamente una consulta que permitirá mostrar un listado de pacientes.

**Salida.-** En la pantalla se mostrará una tabla con todos los pacientes que se encuentren en el listado seleccionado ya que estos estarán organizados por fecha de registro.

#### **Req (03) consulta.recuperación.conjunto Prioridad: Alta**

**Introducción.-** El sistema permitirá la recuperación de conjuntos de datos en atenciones a partir de un solo atributo, como nombre, medico, especialidad, etc.

**Entrada.-** Atributos como ser nombre, medico, especialidad, etc.

**Proceso.-** Se recupera un conjunto de datos en un conjunto.

**Salida.-** Se muestra un conjunto de datos en un conjunto.

#### **Req (04) consulta.recuperación.listado Prioridad: Alta**

**Introducción.-** El sistema permitirá la presentación en pantalla de un listado simple de los registros médicos. Los atributos a mostrar por cada inventario podrán ser elegidos por el usuario.

**Entrada.-** Código de registro.

**Proceso.-** Se realiza un listado de los registros ordenados por código de registro.

**Salida.-** Muestra por pantalla un listado de registros medicos.

#### **Req (05) consulta.recuperación.particiones Prioridad: Alta**

**Introducción.-** El sistema permitirá consultar la lista de los médicos o especialidades que sea solicitado en determinadas atenciones médicas. Para ello, el usuario podrá seleccionar médico, especialidad, fecha. El usuario podrá consultar acerca de las especialidades existentes en almacén o solicitados por un paciente. Adicionalmente, el usuario podrá consultar sobre las especialidades nuevas y cuál es el estado de cada uno de ellos o para proceder a la atención de los mismos.

**Entrada.-** Código de especialidad medica

**Proceso.-** Se busca la especialidad que el paciente está buscando, enviando una respuesta de éxito o rechazo.

**Salida.-** Si la especialidad solicitado se encuentra en el registro se muestra un mensaje de "DISPONIBLE", caso contrario se muestra un mensaje de "NO DISPONIBLE".

#### **Req (06) consulta.navegación Prioridad: Alta**

**Introducción.-** En las consultas de conjunto, el usuario siempre podrá señalar a uno de los médicos mostrados para ver con más detalle sus atributos.

**Entrada.-** Categoría y código de medico seleccionado.

**Proceso.-** La aplicación abrirá el correspondiente vínculo que hace referencia a las características y al médico seleccionado dentro de la categoría, todas las características del médico serán recuperadas del correspondiente depositario.

**Salida.-** En la pantalla del paciente aparecerán todos los detalles del medico seleccionado.

#### **Gestión de Usuarios**

##### **Req (07) gestión. Usuarios Prioridad: Alta**

**Introducción.-** El administrador podrá realizar altas, bajas y modificaciones de los usuarios de la clínica medica, como así también de pacientes.

**Entrada.-** Nombre, ap, am, etc.

**Proceso.-** La primera parte del registro realizará una comprobación de todos los campos del formulario a rellenar, que no sean espacios en blanco, que los campos no estén vacíos, comprobación del correo y clave, la segunda parte realizará la verificación del nombre de usuario, para evitar duplicidades, se existiera se lanzará un mensaje de advertencia, para finalmente insertar una tupla, con estos datos en nuestra base de datos.

**Salida.-** Un mensaje con la leyenda "Inserción de datos efectuados con éxito".

#### **Función de Realizar Citas Medicas**

##### **Req (08) realizar.citasmedicas Prioridad: Alta**

**Introducción.-** Desde esta función se podrá realizar una orden de citas medicas de las especialidades seleccionadas, siempre que el usuario esté registrado.

**Entrada.-** El usuario selecciona una serie de especialidades.

**Proceso.-** El usuario podrá ir añadiendo especialidades a su orden de registro, será en el momento final de la orden cuando se verificará que el usuario esté dado de alta, en caso de que no sea así, el sistema le enviará a una pantalla para que pueda hacerlo, o bien le permita registrarse.

**Salida.-** Aparecerá en la pantalla del paciente un mensaje de pedido realizado de forma correcta, asignando un número de registro.

### **Gestión Medico**

#### **Req (09) Añadir.medico Prioridad: Alta**

**Introducción.-** El administrador a través de un formulario podrá introducir nuevos médicos dentro del sistema.

**Entrada.-** Una nueva línea con el nombre del medico.

**Proceso.-** Una nueva entrada en la base de datos correspondiente con el nuevo medico.

**Salida.-** Una nueva página con un mensaje de éxito.

#### **Req (10) dar.baja.medico Prioridad: Alta**

**Introducción.-** Mediante un formulario el administrador podrá dar de baja a medicos del sistema

**Entrada.-** Debemos seleccionar el medico a dar de baja.

**Proceso.-** El administrador deberá seleccionar un médico a dar de baja dentro del listado existente, para después confirmar pulsando el botón dar de baja.

**Salida.-** Se mostrará una página de éxito si la operación se ha llevado a cabo satisfactoriamente.

#### **Req (11) listar.medico Prioridad: Alta**

**Introducción.-** El administrador podrá efectuar una consulta de todas los médicos.

**Entrada.-** Los valores para realizar la búsqueda.

**Proceso.-** Se efectuará una consulta en la tabla correspondiente a los médicos.

**Salida.-** Se obtendrá en pantalla un listado de todas los médicos actuales.

### **Gestión Especialidad**

#### **Req (12) Añadir.especialidad Prioridad: Alta**

**Introducción.-** El administrador a través de un formulario podrá introducir nuevas especialidades dentro del sistema.

**Entrada.-** Una nueva línea con el nombre del especialidades.

**Proceso.-** Una nueva entrada en la base de datos correspondiente con la nueva especialidad.

**Salida.-** Una nueva página con un mensaje de éxito.

#### **Req (13) dar.baja.especialidad Prioridad: Alta**

**Introducción.-** Mediante un formulario el administrador podrá dar de baja a especialidades del sistema

**Entrada.-** Debemos seleccionar la especialidad a dar de baja.

**Proceso.-** El administrador deberá seleccionar una especialidad a dar de baja dentro del listado existente, para después confirmar pulsando el botón dar de baja.

**Salida.-** Se mostrará una página de éxito si la operación se ha llevado a cabo satisfactoriamente.

#### **Req (14) listar.especialidad Prioridad: Alta**

**Introducción.-** El administrador podrá efectuar una consulta de todas las especialidades.

**Entrada.-** Los valores para realizar la búsqueda.

**Proceso.-** Se efectuará una consulta en la tabla correspondiente a las especialidades.

**Salida.-** Se obtendrá en pantalla un listado de todas las especialidades actuales.

## **Gestión de Salas**

### **Req (15) añadir.salas Prioridad: Alta**

**Introducción.-** Igual que en las anteriores opciones, el administrador a través de un formulario tendrá la opción de añadir una nueva sala dentro de una categoría del catálogo existente.

**Entrada.-** Todas las características de la sala, nombre, especialidad, precio, etc.

**Proceso.-** Una vez introducidos los datos, realizaremos la inserción dentro de nuestra base de datos de salas.

**Salida.-** Una página mostrando el éxito si se ha podido realizar la operación.

### **Req (16) baja.sala Prioridad: Alta**

**Introducción.-** El administrador podrá efectuar la baja de salas del sistema.

**Entrada.-** Deberemos seleccionar la sala a dar de baja.

**Proceso.-** Una vez seleccionada la opción de dar de baja producto, deberemos seleccionar la sala a dar de baja del listado existente. Finalmente confirmaremos la operación con el botón dar de baja.

**Salida.-** Nuevamente aparecerá una página con un mensaje de operación realizada de forma correcta si se ha podido llevar a cabo.

### **Req (17) listar.salas Prioridad: Alta**

**Introducción.-** El administrador podrá efectuar una consulta de todas las salas que estén dados de alta.

**Entrada.-** Los valores para realizar la búsqueda.

**Proceso.-** Se efectuará una consulta en la tabla de salas.

**Salida.-** Se obtendrá en pantalla un listado de todos las salas que se dispongan en la tabla de salas.

### **Req (18) listar.usuarios.medicos Prioridad: Alta**

**Introducción.-** El administrador podrá efectuar una consulta de todos los usuarios y pacientes que estén registrados.

**Entrada.-** Los valores para realizar la búsqueda.

**Proceso.-** Se efectuará una consulta en la tabla correspondiente a los usuarios y pacientes registrados.

**Salida.-** Se obtendrá en pantalla un listado de todos los usuarios y pacientes registrados.

#### **Req (19) cambiar.estado.usuario Prioridad: Alta**

**Introducción.-** El administrador una vez comprobado el listado de usuarios, podrá cambiar el estado de los usuarios.

**Entrada.-** Información del estado de usuarios, mediante consulta cuadro de usuarios.

**Proceso.-** Una vez el administrador verifique que se han efectuado el registro, por el importe de los usuarios solicitados, listará los usuarios y realizará el cambio de estado a baja o activo.

**Salida.-** Volverá a la página principal de administración.

#### **Comunicación bidireccional entre usuarios de la clínica medica y registro de citas medicas**

#### **Req (20) listas.elaboración Prioridad: Baja**

El usuario podrá elegir una serie de usuarios de la clínica con objeto de extraer sus datos personales y los registros que ese usuario hizo efectivas. La elección de dichos usuarios de la clínica y los registros se podrá hacer de dos formas: (1) a partir de una consulta como la descrita en Req (03); (2) manualmente, a partir de una lista de usuarios de la clínica y citas médicas.

#### **Req (21) beneficiario.elaboración Prioridad: Baja**

El sistema deberá dar soporte a la elaboración de la lista de los pacientes beneficiados por la clinica.

#### **Req (22) beneficiario.recepción Prioridad: Baja**

Idealmente, mediante las listas, la dirigencia podrá administrar los beneficios a entregar de cada paciente que quedará almacenar en una BD complementaria a la BD del sistema, pero enlazada con ella.

**Req (23) beneficiario.consulta Prioridad: Media**

El sistema permitirá consultar con facilidad a los pacientes beneficiarios para controlar los beneficios que brinda a sus mejores pacientes, así como los beneficiados que todavía no se registraron en el sistema.

**Elaboración de Estadísticas**

**Req (24) estadísticas.extracción Prioridad: Alta**

Periódicamente, el administrador accederá a la BD y extraerá información con objeto de ser analizada mediante algún paquete estadístico.

Este requisito, en principio, no requiere de ningún esfuerzo de desarrollo, pues toda la BD permite que un administrador extraiga de ella la información necesaria.

**Req (25) estadísticas.extracción.automatización Prioridad: Media**

Con el tiempo, si se considera que ciertos análisis estadísticos son habituales, se considerará la automatización de los mismos, de forma que incluso los usuarios o administradores puedan acceder a los mismos.

**3.3 Requerimientos no funcionales**

**3.3.1 Requisitos de rendimiento**

- La carga de los resultados de búsqueda o listado de especialidades debe ser rápida

**3.3.2 Seguridad**

Cuando un usuario intente conectarse al sistema deberá introducir su nombre de usuario y clave de acceso y el sistema deberá comprobar que se trata de un usuario autorizado.

El sistema de información tendrá distintos tipos de usuarios y a cada uno de ellos se les permitirá únicamente el acceso a las funciones que le correspondan.

El sistema proporcionará además funciones de auditoría, registrando la fecha, hora y usuario de toda transacción.



El sistema no permitirá el cambio de datos de manera arbitraria, no permitirá cambios en los precios de las citas médicas ya registradas en el sistema.

Es importante disponer de un sistema de backups automatizado, que permita volver a un estado anterior tras un desastre informático.

El procedimiento de recuperación luego de una caída debe estar documentado.

### **3.3.3 Fiabilidad**

Cualquier transacción finalizada por un usuario deberá ser procesada exitosamente con una tasa de error de 0%.

### **3.3.4 Disponibilidad**

El sistema deberá proveer tolerancia a fallos garantizando una disponibilidad del 99,99%.

El sistema estará disponible las 24 horas del día, los 7 días de la semana.

### **3.3.5 Mantenibilidad**

El sistema requerirá el mínimo posible de mantenimiento, siempre que sea posible, las tareas de mantenimiento deberían realizarse por el equipo de desarrollo.

### **3.3.6 Portabilidad**

Todos los modelos deberán ser independientes de cualquier plataforma.

## **3.4 Otros requisitos**

En cuanto a la infraestructura informática necesaria se requiere una interfaz web con un Sistema de Gestión de Base de Datos. La primera opción que viene es PostgreSQL, pero se podrían evaluar otras alternativas.

# PRUEBAS DE SOFTWARE

## CAJA BLANCA Y CAJA NEGRA

*IMPLEMENTACION DE HERRAMIENTAS PARA LA VERIFICACION ERRORES  
ENEL SISTEMA CLINICA "EL BUEN SAMARITANO"*

ESTUDIANTE: POSADAS ALANEZ ULISES YAMIL  
TARIJA - BOLIVIA

# TABLA DE CONTENIDO

## Contenido

Introduccion_____	1
Resumen del Proyecto_____	2
Tipos de Prueba_____	2
Objetivo de la Prueba_____	3
Herramienta Requerida_____	4
Características de la Herramienta_____	4
aplicación de Casos de Pruebas_____	5

# CLINICA EL BUEN SAMARITANO

## 1. PROYECTO DE SOFTWARE

### 1.1 INTRODUCCION

La base de las pruebas de un software son las más costosas del ciclo de la vida del software. Es sentido estricto, deben realizarse pruebas de todo lo generado durante la construcción del producto, lo que incluye especificación de requisitos, casos de uso, diagramas de diversos tipos y por su puesto el código fuente.

El ingeniero crea una serie de casos de prueba que intentan "demoler" el software que ha sido construido. Tiene como objetivos:

- 1.- La prueba es el proceso de ejecución de una programación la intención de descubrir error.
- 2.- Un buen caso de prueba es aquel que tiene una alta probabilidad de mostrar un error no descubierto hasta entonces.
- 3.- Una prueba tiene éxito si descubre un error no detectado hasta entonces.

### 1.2 Resumen del proyecto

El proyecto es capaz de automatizar la gestión y administración de los datos que existen en un Hotel cualquiera, permitiéndose registrar personal, pacientes, realizar citas médicas y asignar roles para la administración del sistema entre otras.

## 2. TIPOS DE PRUEBAS

### 2.1 TEST DE PRUEBA CAJA BLANCA CON "SONNAR CUBE"

Es una plataforma para evaluar código fuente. Es software libre y usa diversas herramientas de análisis estático de código fuente como Checkstyle, PMD o FindBugs para obtener métricas que pueden ayudar a mejorar la calidad del código de un programa

### HERRAMIENTA REQUERIDA

StartSonar.bat

Comando con el cual se ejecuta un servidor que nos mostrara el respectivo test que se realiza al código de nuestro proyecto.

sonar-runner

Este comando realiza la revisión de nuestro proyecto clase por clase y genera un informe detallado que se mostrara en el servidor web de Sona@ube.

# CLINICA EL BUEN SAMARITANO

## APLICACIÓN DE CASOS DE PRUEBA CON "SONARQUBE"

*Una vez ejecutado nuestro servidor SonarQube*

The first screenshot shows a Windows command prompt running the SonarScanner.bat file. The output displays various configuration details, including the project root, SonarQube Scanner version (2.8), Java version (1.8.0\_65), and the location of the SonarQube server (localhost:9000). The second screenshot shows the SonarScanner.bat file running again, displaying the results of the analysis, including the number of lines of code (LOC), bugs, vulnerabilities, and the last analysis date (21/Nov/2016). The third screenshot shows the SonarQube web interface, displaying the same analysis results in a table format.

Version	LOC	Bugs	Vulnerabilities	Code Smells	Last Analysis
1.0	31	0	0	10	21/Nov/2016
1.3	0	0	0	0	18:01

Procedemos a ir a la página web del servidor SonarQube

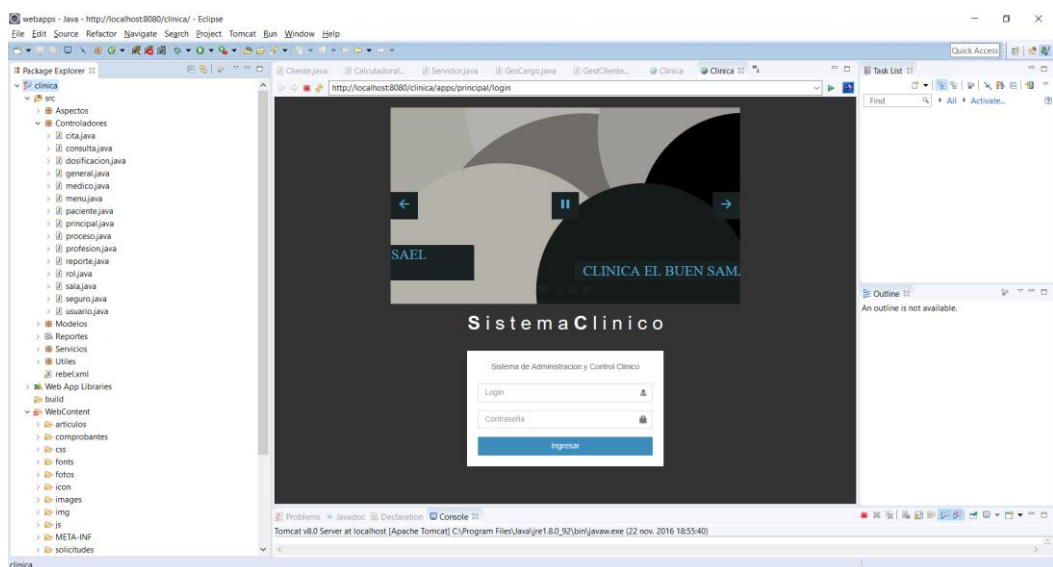
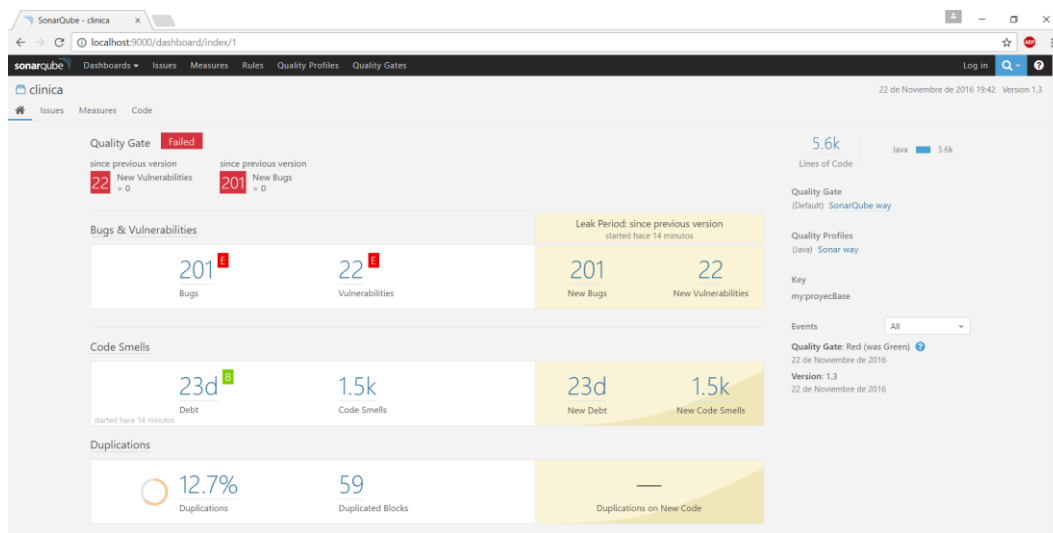
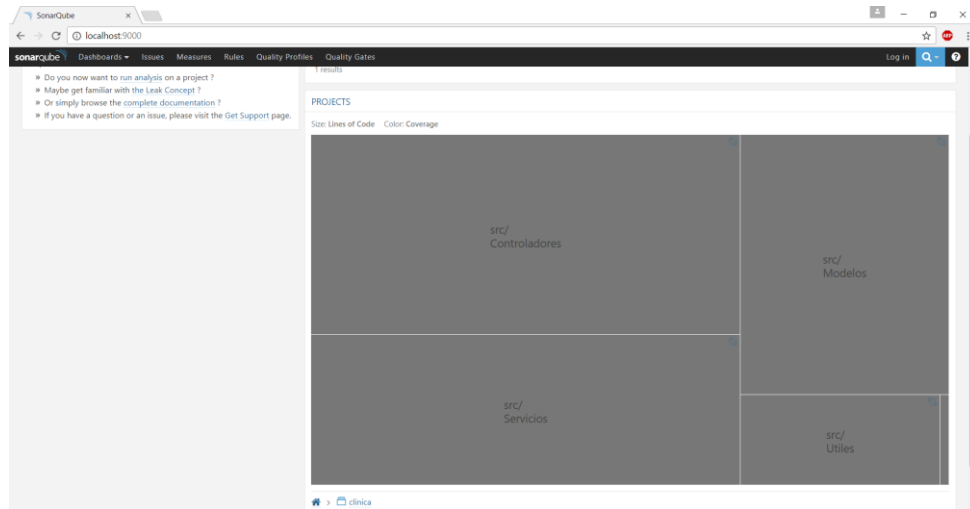
<http://localhost:9000/>

El cual nos muestra a detalle los módulos que fueron revisados.

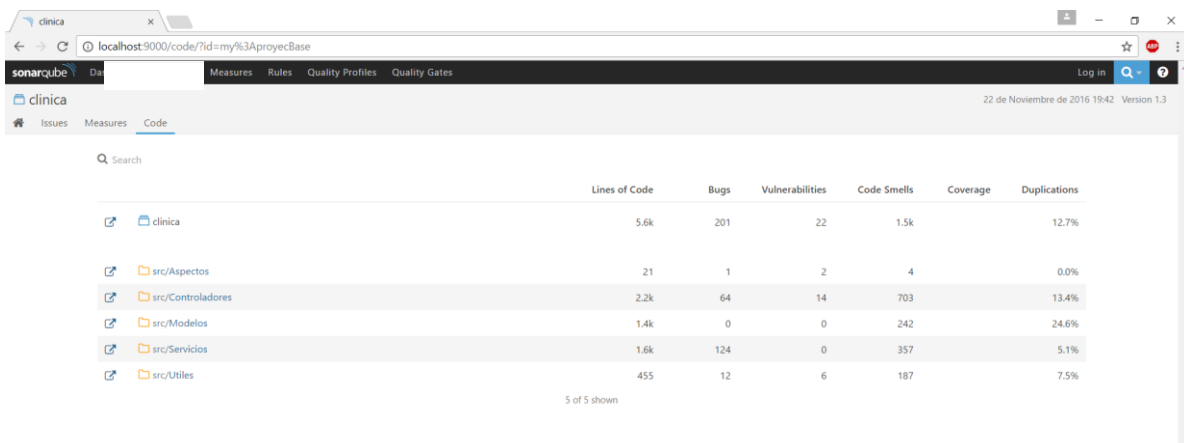
En este caso se realizó el test a los controladores del proyecto Clínica

# CLINICA EL BUEN SAMARITANO

## "El Buen Samaritano"



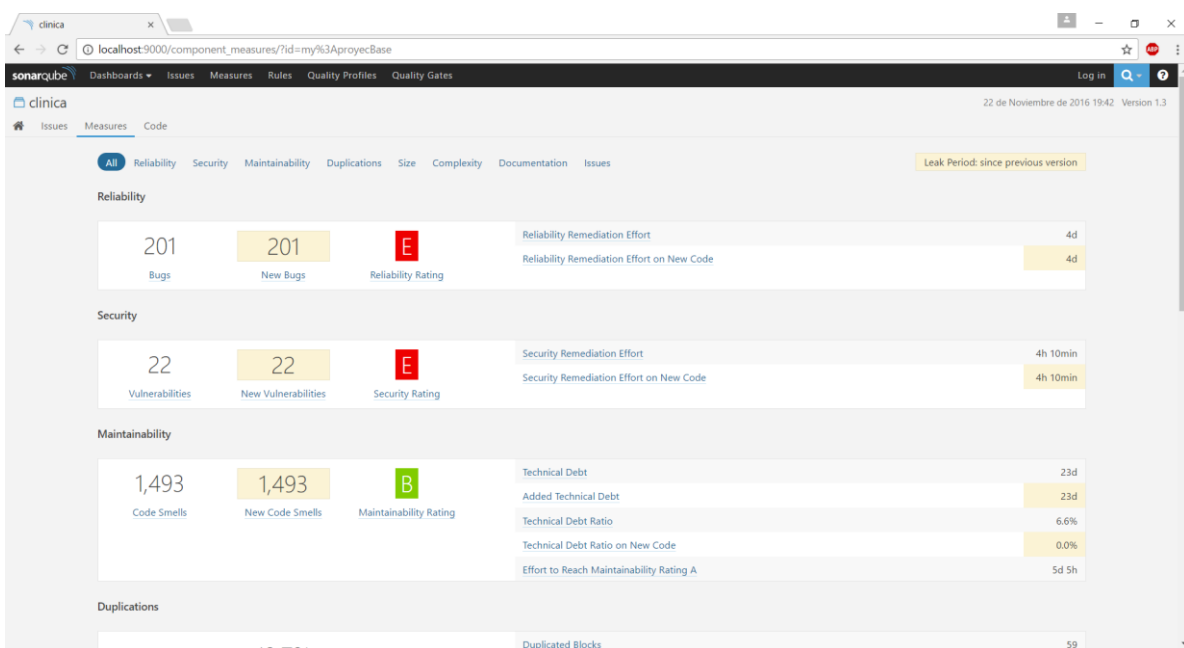
# CLINICA EL BUEN SAMARITANO



The screenshot shows the SonarQube Code Quality Dashboard for the 'clinica' project. The dashboard displays various quality metrics across different components of the project.

Component	Lines of Code	Bugs	Vulnerabilities	Code Smells	Coverage	Duplications
clinica	5.6k	201	22	1.5k		12.7%
src/Aspectos	21	1	2	4		0.0%
src/Controladores	2.2k	64	14	703		13.4%
src/Modelos	1.4k	0	0	242		24.6%
src/Servicios	1.6k	124	0	357		5.1%
src/Utilis	455	12	6	187		7.5%

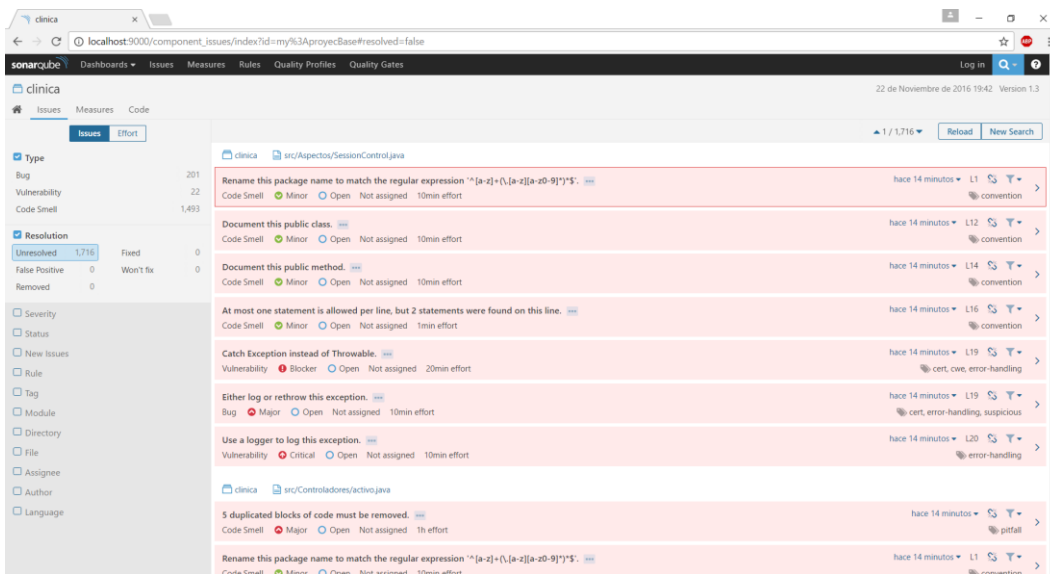
5 of 5 shown



The screenshot shows the SonarQube Component Measures Dashboard for the 'clinica' project. The dashboard displays various quality metrics across different components of the project.

Category	Value	Unit	Rating	Remediation Effort
Reliability	201	Bugs	E	4d
Security	22	Vulnerabilities	E	4h 10min
Maintainability	1,493	Code Smells	B	23d
Duplications	59	Duplicated Blocks		

## Errores y Correcciones por orden de Gestión

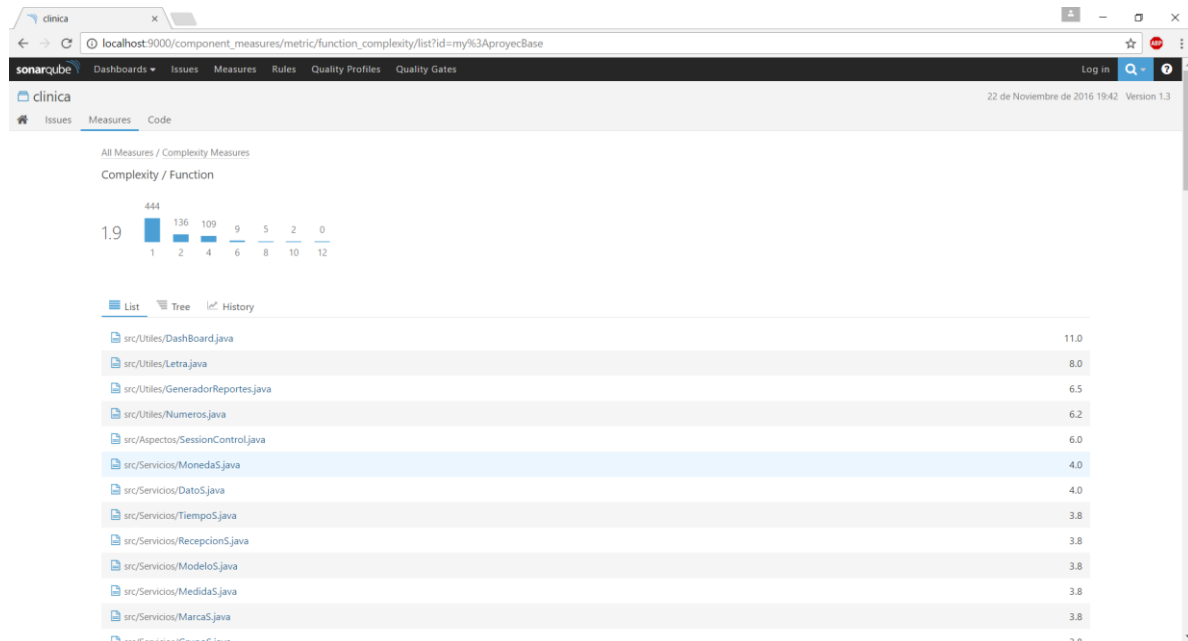


The screenshot shows the SonarQube Issues Dashboard for the 'clinica' project. The dashboard displays various quality metrics across different components of the project.

Issue	Severity	Resolution	Effort
Rename this package name to match the regular expression "[a-z]+([a-z][a-z0-9]*)\$"	Minor	Open	10min effort
Document this public class.	Minor	Open	10min effort
Document this public method.	Minor	Open	10min effort
At most one statement is allowed per line, but 2 statements were found on this line.	Minor	Open	1min effort
Catch Exception instead of Throwable.	Minor	Open	20min effort
Either log or rethrow this exception.	Minor	Open	10min effort
Use a logger to log this exception.	Minor	Open	10min effort
5 duplicated blocks of code must be removed.	Major	Open	1h effort
Rename this package name to match the regular expression "[a-z]+([a-z][a-z0-9]*)\$"	Minor	Open	10min effort

# CLINICA EL BUEN SAMARITANO

## Complejidad por Archivo del Proyecto Clínica



## 2.2 PRUEBAS DE CAJA NEGRA

Casos de Prueba 1	Selección de las opciones presentadas en la pantalla "Gestión Usuario"
Propósito	Comprobar que esta habilitando las opciones presentadas en la pantalla de "Gestión Usuario"
Pre Requisitos	El Usuario debe haber iniciado sesión en el sistema.
Casos de Entrada	<ul style="list-style-type: none"><li>Filtro botón "Buscar"</li><li>Botón "Nueva Usuario"</li><li>Boton "Modificar Usuario"</li><li>Botón "Eliminar Usuario"</li></ul> Entradas 4 Salidas 1
Pasos	<ul style="list-style-type: none"><li>Ingresar nombres o palabras claves para buscar rápidamente cualquier Usuario</li><li>Hacer clic en "Buscar"</li><li>Hacer clic en "Nuevo Usuario"</li><li>Hacer clic en "Modificar Usuario"</li><li>Hacer Clic en "Eliminar Usuario"</li></ul>
Resultado Esperado	<ol style="list-style-type: none"><li>Al hacer clic en buscar aparecerá una nueva ventana deseada si es que existe la misma.</li><li>Enviara al usuario a la pantalla Añadir nuevo Usuario</li></ol>



# CLINICA EL BUEN SAMARITANO

	<ol style="list-style-type: none"> <li>3. Enviar al usuario a la pantalla Modificar Usuario</li> <li>4. Eliminará los datos del Usuario seleccionado de la lista que aparece en la pantalla</li> </ol>
--	--

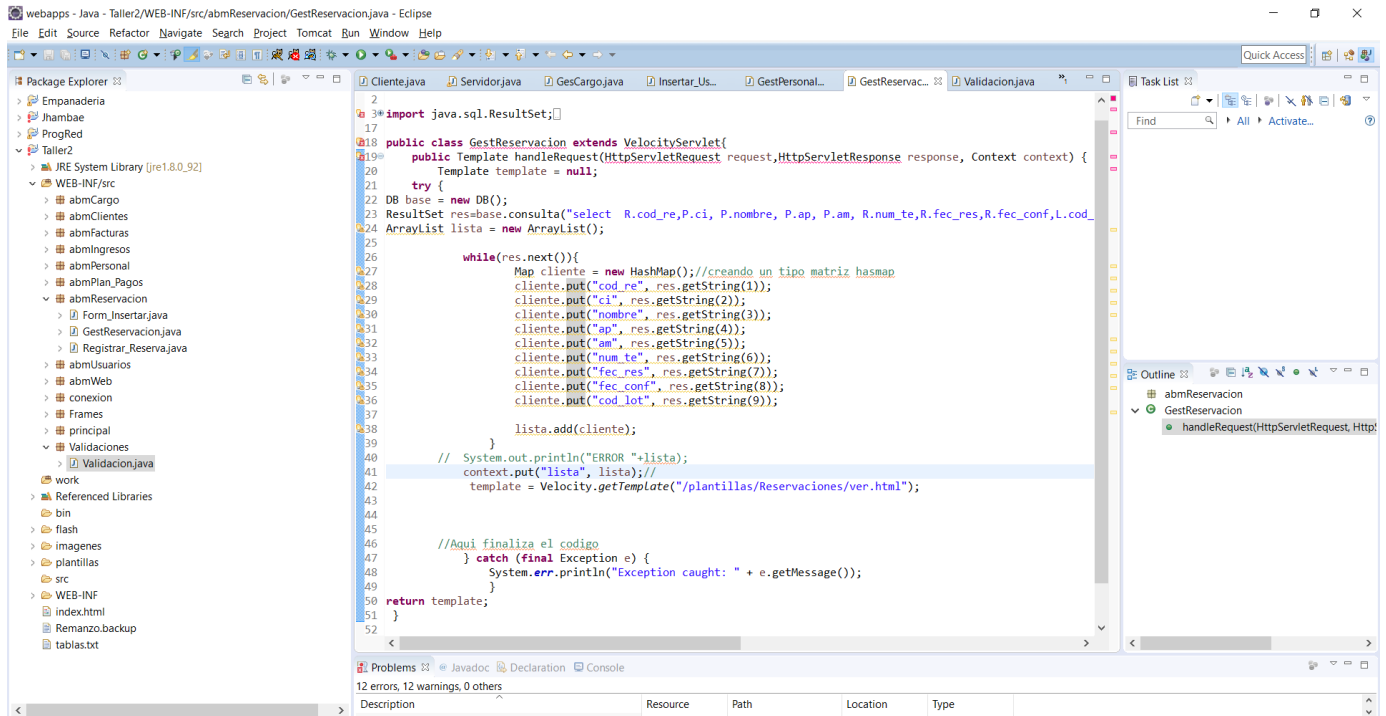
Casos de Prueba 2	Selección de las opciones presentadas en la pantalla "Gestión Paciente"
Proposito	Comprobar que esta habilitando las opciones presentadas en la pantalla de "Gestión Paciente"
Pre Requisitos	El Usuario debe haber iniciado sesión en el sistema.
Casos de Entrada	<ul style="list-style-type: none"> <li>• Filtro botón "Buscar"</li> <li>• Botón "Nueva Reserva"</li> <li>• Botón "Borrar"</li> </ul> Entradas 3 Salidas 1
Pasos	<ul style="list-style-type: none"> <li>• Ingresar nombres o palabras claves para buscar rápidamente cualquier venta</li> <li>• Hacer clic en "Buscar"</li> <li>• Hacer clic en "Nueva Reserva"</li> <li>• Hacer Clic en "Borrar"</li> </ul>
Resultado Esperado	<ol style="list-style-type: none"> <li>5. Al hacer clic en buscar aparecerá una nueva ventana deseada si es que existe la misma.</li> <li>6. Enviara al usuario a la pantalla Añadir nueva Reserva</li> <li>7. Eliminará la Reserva seleccionada de la lista que aparece en la pantalla</li> </ol>

## APLICACIÓN DE CASOS DE PRUEBA CON "CRAP4J"

Una vez instalado u ejecutado "Crap4j" en Eclipse, nos generará un reporte que podremos ver en un navegador que se abrirá automáticamente cuando ejecutamos "Crap4j"

El testeo está basado sobre los controladores principales del Proyecto Taller2

# CLINICA EL BUEN SAMARITANO



## AbmReservacion/GestReservacion Código Java

```
package abmReservacion;
```

```
import java.sql.ResultSet;
import java.util.ArrayList;
import java.util.HashMap;
import java.util.Map;
```

```
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
```

```
import org.apache.velocity.Template;
import org.apache.velocity.app.Velocity;
import org.apache.velocity.context.Context;
import org.apache.velocity.servlet.VelocityServlet;
```

```
import conexion.DB;
```

```
public class GestReservacion extends VelocityServlet{
    public Template handleRequest(HttpServletRequest request,HttpServletResponse response, Context context) {
        Template template = null;
        try {
            DB base = new DB();
            ResultSet res=base.consulta("select R.cod_re,P.ci, P.nombre, P.ap, P.am,
            R.num_te,R.fec_res,R.fec_conf,L.cod_lot from persona P ,reserva R, lote L
            where(R.cod_p = P.cod_p ) and R.estado='1' and R.cod_lot=L.cod_lot;");
            ArrayList lista = new ArrayList();

            while(res.next()){
                Map cliente = new HashMap();//creando un
                tipo matriz hasmap
                cliente.put("cod_re", res.getString(1));
                cliente.put("ci", res.getString(2));
```

```

        cliente.put("nombre", res.getString(3));
        cliente.put("ap", res.getString(4));
        cliente.put("am", res.getString(5));
        cliente.put("num_te", res.getString(6));
        cliente.put("fec_res", res.getString(7));
        cliente.put("fec_conf", res.getString(8));

        cliente.put("cod_lot", res.getString(9));

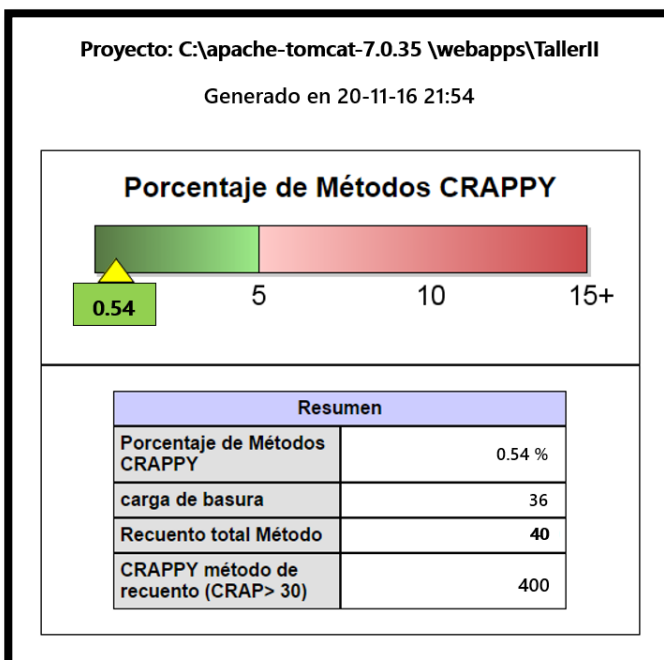
        lista.add(cliente);
    }
    // System.out.println("ERROR "+lista);
    context.put("lista", lista);
    template =
Velocity.getTemplate("/plantillas/Reservaciones/ver.html");
    //Aqui finaliza el codigo
    } catch (final Exception e) {
        System.err.println("Exception caught: " +
e.getMessage());
    }
    return template;
}

}

```

## Generador de reportes en Navegador web

Automáticamente de abre un navegador el cual no mostrara el testeo realizado por Crap4f, el cual data un detalle de los errores encontrados.

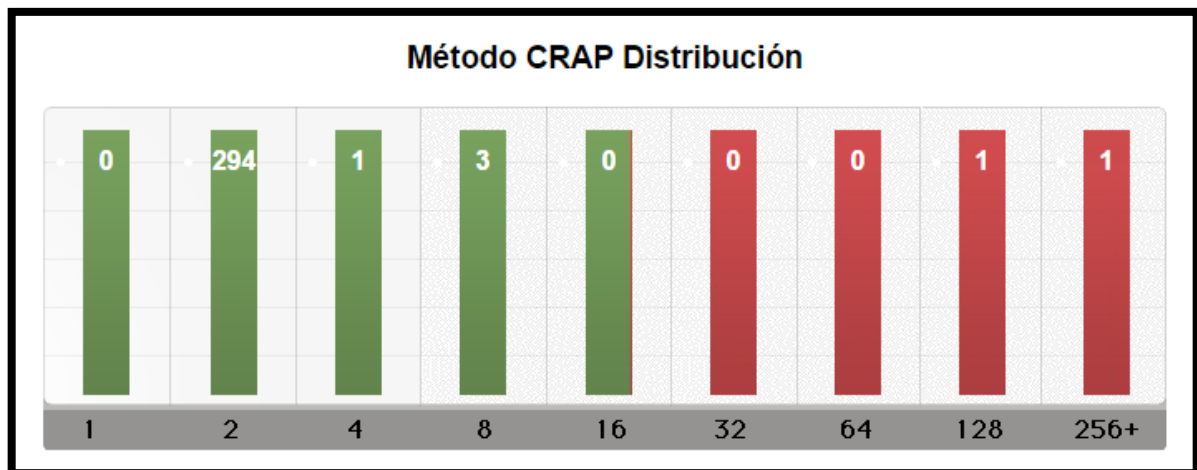


## Test realizado al Proyecto TallerII

- **Porcentaje de Métodos con error** 0.54%
- **Carga de Basura** 36
- **Recuento total de métodos** 400

Cuadro de Barras el cual nos muestra valor general del testeo al Proyecto TallerII

# CLINICA EL BUEN SAMARITANO



La barra en rojo nos muestra posibles errores en el sistema

Cuadro en el cual se muestra un detalle exhaustivo de todos los métodos del sistema TallerII los cuales fueron Testeados

Detalle:

- Método Testeado
- Complejidad
- Cobertura del Código
- Residuos del sistema "basura"
- Carga de residuos "basura"

Project: C:\apache-tomcat-7.0.35\webapps\TallerII

[Overview Page](#) | [CRAP](#) | [Complexity](#) | [Coverage](#)

Method	Complexity	Coverage	CRAP	CRAP Load
public static java.lang.String ultimodiaMes(int, int) Utilidades . fechas	17	0,00 %	306,00	17
public static java.lang.String obtenerMesLiteral(int) Utilidades . fechas	13	0,00 %	182,00	13
public void <init>() Aspectos . SesionCerrada	1	0,00 %	2,00	0
public java.lang.String logeo(org.springframework.ui.Model, HttpServletRequest) Aspectos . SesionCerrada	1	0,00 %	2,00	0
public void <init>() Utilidades . ContadorMenu	1	0,00 %	2,00	0
public java.lang.Integer getContCodm() Utilidades . ContadorMenu	1	0,00 %	2,00	0
public void setContCodm(java.lang.Integer) Utilidades . ContadorMenu	1	0,00 %	2,00	0
public void <init>() Utilidades . Utilitarios	1	0,00 %	2,00	0
public java.lang.String dateFormat(java.util.Date) Utilidades . Utilitarios	1	0,00 %	2,00	0
public java.lang.String decimalFormat(java.lang.String, double) Utilidades . Utilitarios	1	0,00 %	2,00	0
public void <init>() Utilidades . ex	1	0,00 %	2,00	0
public static java.lang.String decodificar(java.lang.String) Utilidades . ex	1	0,00 %	2,00	0

## 2.3 TEST DE PRUEBA CON "CHECKSTYLE"

# CLINICA EL BUEN SAMARITANO

Checkstyle es una herramienta de desarrollo que ayuda a los programadores a escribir código Java adecuándose a estándares de codificación establecidos, facilitando para ello la automatización del proceso de chequeo del código generado.

Checkstyle incorpora las recomendaciones de Sun sobre el estilo de código, pero estas reglas pueden ser redefinidas e incluso creadas completamente desde cero por el usuario, lo que convierte este plug-in en adaptable al estilo de codificación interno de nuestro entorno, sea cual sea.

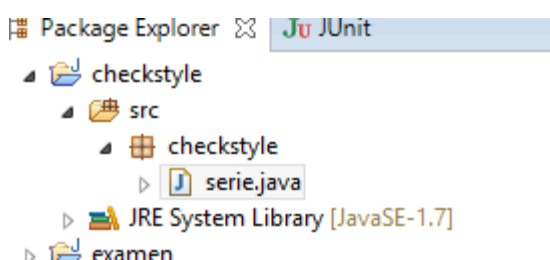
## HERRAMIENTAS DE INTEGRACIÓN CON IDES

Checkstyle se integra a varios IDEs a través de distintos plugins de terceros.

Algunos de los más conocidos son:

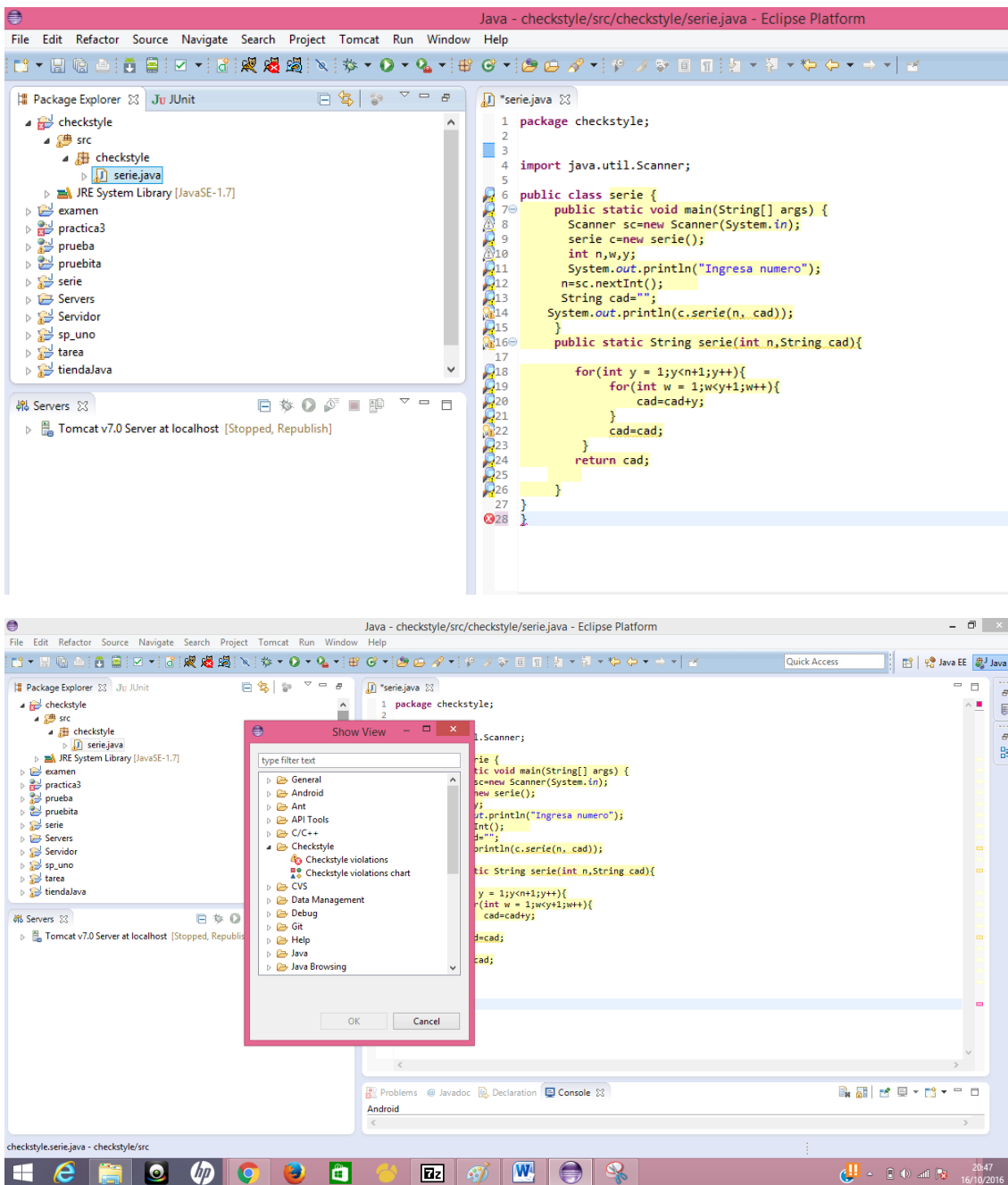
- [Eclipse-CS](#)
- [Checkclipse](#)
- [Checkstyle-idea](#)
- [JetStyle](#)
- [Checkstyle Beans](#)
- [nbCheckStyle](#)
- [JBCS](#)
- [jbCheckStyle](#)

La clase a testear con la herramienta "checkstyle" es el logueo el cual pertenece al paquete de controladores principales del sistema checkstyle  
clase serie



10.- luego vamos a probar como vemos hay errores

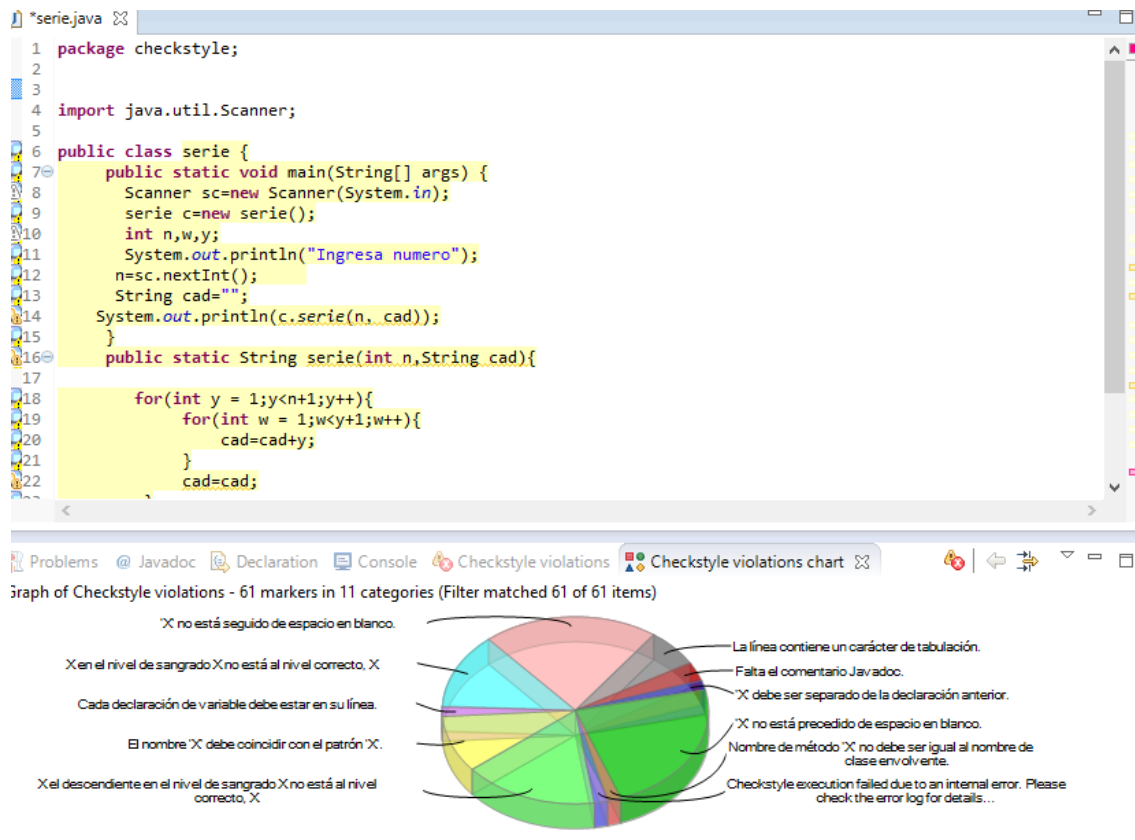
# CLINICA EL BUEN SAMARITANO



## Resultados

\*Checkstyle violations chart

# CLINICA EL BUEN SAMARITANO



\*checkstyle violations